Bidirectional Appearance Distribution Function for Stylized Shading

David Vanderhaeghe, Romain Vergne, Pascal Barla, William Baxter

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Bidirectional Appearance Distribution Function for Stylized Shading

David Vanderhaeghe — Romain Vergne — Pascal Barla — Bill Baxter

N° 7195

Février 2010

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--7195--FR+ENG

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

INRIA

centre de recherche BORDEAUX - SUD OUEST

Bidirectional Appearance Distribution Function for Stylized Shading

David Vanderhaeghe*, Romain Vergne*, Pascal Barla*, Bill Baxter†

Thème : Perception, cognition, interaction Interaction et visualisation
Équipe-Projet IPARLA

Rapport de recherche n° 7195 — Février 2010 — 16 pages

Abstract: We define a new shading tool called a Bidirectional Appearance Distribution Function (BADF) tailored to the direct control of stylized appearance. A BADF can be thought of as defining the appearance of a sphere from all possible illumination directions. Our BADF formulation generalizes and improves upon previous stylized shading techniques by enabling the direct control of shading profiles in screen space, exaggerating surface features in a flexible manner, and letting users control stylized appearance from multiple lighting or viewing directions. This allows users to start from a simple shading behavior, and refine from there towards greater stylization. Our GPU implementation works in real-time, which benefits both editing, and rendering in interactive systems. These features make BADFs an efficient tool for many applications in artistic and scientific illustration domains.

Key-words: Shading, Expressive Rendering, NPR

* Iparla – LaBRI – Université de Bordeaux/INRIA Bordeaux – Sud-Ouest
† Microsoft Research

Centre de recherche INRIA Bordeaux – Sud Ouest
Domaine Universitaire - 351, cours de la Libération 33405 Talence Cedex
Téléphone : +33 5 40 00 69 00

Fonction de distribution bidirectionnelle de l'apparence pour l'éclairement stylisé

Résumé : Nous définissons un nouvel outil d'éclairement appelé Fonction de distribution bidirectionnelle de l'apparence (BADF). Cet outil est conçu pour le contrôle direct de la stylisation de l'apparence. Une BADF peut être vue comme la définition de l'apparence visuelle d'une sphère pour toutes les directions de lumières possibles. Notre formulation de la BADF généralise et améliore les méthodes de stylisation précédentes car elle permet à l'utilisateur de contrôler directement le profil des primitives d'éclairement en espace écran, d'exagérer les détails de surface de manière flexible et de lui laisser la possibilité de styliser plusieurs directions de vue et/ou d'éclairage. Ceci permet à l'utilisateur de partir d'un rendu classique et de le modifier pour obtenir un éclairement riche et stylisé. Notre implémentation sur carte graphique permet un rendu en temps réel, autorisant l'édition interactive et l'utilisation de notre méthode dans des contextes de rendu temps réel. Les BADFs sont donc un outil efficace pour les applications artistiques et de visualisation scientifique.

Mots-clés : Eclairement, Rendu expressif, NPR

1 Introduction

Shading in Computer Graphics is generally designed to simulate reality by using a combination of complex materials and lighting environments. Although this approach has many applications in architecture or special effects for movies, other applications rather use shading in a stylized manner. In scientific illustrations (e.g., archeology, medicine, botanics, anatomy, mechanics), shading depicts materials and illumination in simplified ways [Woo94, Hod03] to communicate clearly object shape or function. Whereas in artistic illustrations (e.g., comics, anime, graphic novels, illustrated books), conventional shading is radically altered, for instance by creating sharp color transitions that only give a hint of material or illumination. Such highly stylized shading is chosen to set different moods or atmospheres, or exaggerate object structure [Hog91, McC94].

With hand-made stylized shading, material and lighting properties are only implicitly given, and do not need to match physically realistic constraints. These additional degrees of freedom regarding physical accuracy are traditionally used to enhance *object shape* through shading. Indeed, skilled artists are able to reveal smooth or sharp surface features by creating subtle or strong color variations around them. Such flexibility might even be used to design sophisticated *dynamic behaviors*, since shading does not need to be consistent across varying viewing or lighting conditions. In scientific illustrations [Woo94], variations in shading may provide a means to better depict object parts that are revealed under specific viewpoints. In artistic illustrations [Hog91], variations in shading may change the perception of a scene depending on the overall illumination direction.

The main goal of this paper is to provide an approach to stylized shading for both scientific and artistic illustrations. Ideally, users should be able to design stylized shading appearance without having to deal with complex materials or illumination at all. This is the main idea behind the *LitSphere* of Sloan et al. [SMGG01], which is the previous method most similar to our work. It consists of a single shaded sphere stored in a 2D texture and looked up using screen-space normals. An artist simply draws the desired appearance using painting software or captures it from existing paintings. Variations on this basic idea have been successfully used in different contexts such as volumetric rendering [BG07] or digital sculpture software such as ZBrush®.

However, because of its reduced dimensionality and simple look-up function, a LitSphere is limited with regards to our target applications: 1) Diffuse or glossy materials cannot properly be imitated since the illumination orientation is "baked" into the model; 2) Disturbing sliding artifacts may occur when high-frequency variations are found in the texture; 3) Shape features are only enhanced by using special (e.g., darker) colors along contours.

Our main contribution to stylized shading is the introduction of *Bidirectional Appearance Distribution Functions*. A BADF explicitly models the stylized appearance of a shaded sphere under every possible illumination orientation. We show in Section 3.1 that it can be defined as a 4D function that takes a normal and a view direction as input and outputs a color. Increasing dimensionality is the key to address the main limitation of LitSpheres: since multiple illumination orientations are taken into account, a BADF can properly represent the characteristic effects of opaque materials such as diffuse gradients or highlights, and the way they move. Defining such a 4D function with an image-based approach would not be tractable because it would require too much user interaction. Instead, we introduce a generalized Phong model which allows artists to easily achieve a wide range of stylized shading appearances. As explained in Section 3.2, it is able to reproduce Phong, Half-Lambert and cartoon shading as special instances. Our model also permits to depict salient surface details by tweaking shading as detailed in Section 3.3, which would not be possible with an image-based technique. Such an approach enhances shape through shading in ways that more closely resemble hand-made illustrations compared to previous work.

With these features, it takes very few steps to create interesting BADFs. In addition, for more advanced shading behaviors, users may design predefined stylized appearances, and let our model interpolate appearances between key viewing or lighting directions, as explained in Section 3.4. With directional control, the more time spent adding key directions, the richer the resulting stylized appearances can be. We show examples of artistic and scientific illustrations obtained with our technique in Sections 4 and discuss its limitations in Section 5.

2 Previous work

The earliest form of direct appearance control we are aware of is inverse rendering. Many methods have been proposed, some enabling material editing [BAOR06, CPK06, EBJ*06], others for the control of illumination [KPC93, SDS*93, RH01]. The inverse rendering method closest to our approach is the *Illumination Brush* of Okabe et al. [OMSI07]. Starting from a known BRDF, they infer the environment lighting by directly painting on the object the desired result of either diffuse or specular reflections. Even if the paint strokes may be moved around for stylization purpose, the adaptation of the method to the creation of stylized shading is not straightforward, since it relies on BRDF models and realistic illumination and materials. We rather seek a method that is not bound by such physical constraints.

Many stylization techniques have based their approach on a modification of an existing shading model. The simplest one is *Toon Shading*, which simply segments the result of a diffuse shading function into two or three color bands. The *Technical Illustration Shader* of Gooch et al. [GGSC98] uses a similar approach with an unclamped diffuse term (also called *Half-Lambertian*) and cool-to-warm color gradients. It has also been found to offer better depiction of shape in video game applications [MFE07]. The *X-Toon* method of Barla et al. [BTM06] extends the 1D toon function to a 2D function stored in a texture. The additional dimension is used for creating back-lighting or depth-of-field effects, for instance. However, even with local control [TABI07], the range of styles provided by remapping a pseudo-diffuse function to a color gradient is limited.

These shading methods have been used as building blocks in more recent work that focus on shape depiction through shading. The *Exaggerated Shading* of Rusinkiewicz et al. [RBD06] employs a Half-Lambertian and a hierarchy of simplified normals to align the light direction with grazing angles. It creates an accurate depiction of surface details, but is tied to the use of a Half-Lambertian shading. In contrast, the *Apparent Relief* technique of Vergne et al. [VBGS08] is not correlated to a specific shading model, but it only uses shape features as a mask to blend between a pair of shading appearances.

Other techniques modify Phong shading in order to manipulate highlights specifically. The system of Anjyo and Hiramitsu [AH03] provides control over the shape of cartoon highlights: they interpolate parameters of highlight transformations across time. Existing operations include translation, rotation, scaling, splitting, squaring, and boolean operators. The method has been extended [AWB06] to control the location of cartoon diffuse shading (modifying light direction), and to manipulate directly the shading on an object's surface. However, the manipulations are object-specific, and thus do not easily transfer between different 3D objects. A similar approach for highlights has been subsequently proposed by Pacanowski et al. [PGSP08]. Their method is not limited to cartoon highlights, but works for any star-shaped highlight with its appearance defined by a color gradient. They also propose a simple technique for varying appearance depending on a few predefined illumination directions.

Our BADF model encompasses techniques based on Lambertian (including half-Lambertian) or Phong shading models and incorporates surface shape features by applying complex shading variations without any apriori on shading. The resulting stylized appearance is controlled from arbitrary key lighting or viewing directions, and works in real-time on modern graphics hardware. It is also easily transferred between 3D objects, as opposed to methods that require establishing a specific surface parametrization.

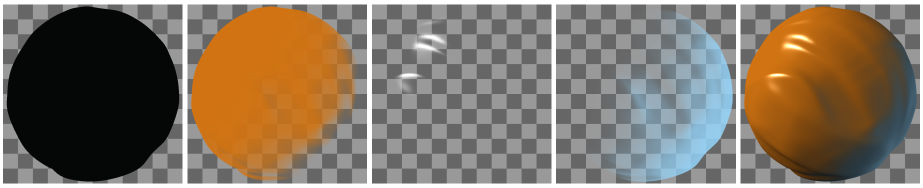

Figure 1: ***Layered BADF model****: each light source is assigned to a layer in a way similar to image editing software. Here we illustrate this organization on a simple BADF. From left to right: a fill layer, a diffuse layer, a specular layer, a back-lighting layer, and the composited result. More complex BADFs are shown in subsequent figures.*

3 BADF model

One of the most important considerations to an artist when designing appearance is the location of the dominant light source [Woo94]. Given this information, artists are able to shade surface regions depending on their orientation relative to that light, and place secondary lights to create more complex effects. Our approach is to define an *illumination reference frame* $\mathcal{L}$, and to define all light sources in this frame. End-users then manipulate $\mathcal{L}$ in the same way they would manipulate an environment illumination map.

3.1 Definitions

The main idea of a BADF is to encapsulate the definition of L directional light sources $\mathbf{l}_i$, their interaction with the object material, and stylistic effects, into a single function. Since all light sources are expressed in $\mathcal{L}$, we also express the normal $\mathbf{n}$ and view $\mathbf{v}$ directions in $\mathcal{L}$ and define a BADF as a 4D function of $\mathbf{n}$ and $\mathbf{v}$:

$$\begin{array}{rccc} \rho_{\mathcal{A}}: & \mathbb{S}^2 \times \mathbb{S}^2 & \rightarrow & [0,1]^4 \\ & \mathbf{n}, \mathbf{v} & \mapsto & (r, g, b, a) \end{array} \quad (1)$$

where $\mathbb{S}^2$ is the sphere of directions.

In this paper, we introduce a *layered* BADF model that takes inspiration from image and video editing software such as Photoshop® or Shake® (see Figure 1 and the supplemental video). Each layer is assigned a light source direction $\mathbf{l}_i$, a shading function ρ_i and a color $\mathbf{c}_i$. The BADF is obtained by combining layers together with user-specified blending functions $\odot_i$, one per layer. We use additive, multiplicative, subtractive and alpha blending functions in our system. More formally, our BADF model is given by:

$$\begin{array}{rcl} \rho_{\mathcal{A}}(\mathbf{n}, \mathbf{v}) & = & \mathbf{C}_L(\mathbf{n}, \mathbf{v}) \\ \mathbf{C}_i(\mathbf{n}, \mathbf{v}) & = & (\rho_i(\mathbf{n}, \mathbf{v})\ \mathbf{c}_i) \odot_i \mathbf{C}_{i-1}(\mathbf{n}, \mathbf{v}) \end{array} \quad (2)$$

where $\mathbf{C}_0(\mathbf{n}, \mathbf{v})$ is a constant background color and $i \in [1, L]$. If, for all i, we choose $\odot_i$ to be the additive blending function and $\rho_i(\mathbf{n}, \mathbf{v}) = \rho(\mathbf{l}_i, \mathbf{v})(\mathbf{n} \cdot \mathbf{l}_i)$ with ρ a Bidirectional Reflectance Distribution Function (BRDF), then we fall back to the "conventional" shading model.

In stylized shading though, it is more interesting for artists to try out different combinations of layers, and to make use of shading functions that depart from realistic BRDFs. Our BADF model provides such flexibility via three key technical contributions: a shading function (Section 3.2) directly controllable in image-space via isocurve widgets; a warping method (Section 3.3) that enhances surface features by manipulating layer color and shading function; and a directional interpolation technique (Section 3.4) that varies layer parameters based on key lighting or viewing directions. When used in combination, these techniques provide for a wide range of stylized appearances, as shown in Figures 6 and 7.

3.2 Shading function

In our model, each of the L directional light sources not only conveys material effects such as diffuse gradients or highlights, but also stylistic effects such as sharp, cartoon-like transitions, or Half-Lambert fall-offs. This is obtained via the shading function ρ_i, defined as the composition of an angular parametrization $u_i : \mathbb{S}^2 \times \mathbb{S}^2 \rightarrow [0, \pi]$ and a profile function $\alpha_i : [0, \pi] \rightarrow [0, 1]$:

$$\rho_i(\mathbf{n}, \mathbf{v}) = \alpha_i \circ u_i(\mathbf{n}, \mathbf{v}). \quad (3)$$

The choice of parametrization u_i controls the way shading moves onto the surface and is defined relative to the lighting direction $\mathbf{l}_i$:

$$u_i(\mathbf{n}, \mathbf{v}) = \operatorname{acos}(\mathbf{d}_i \cdot \mathbf{l}_i) \quad (4)$$

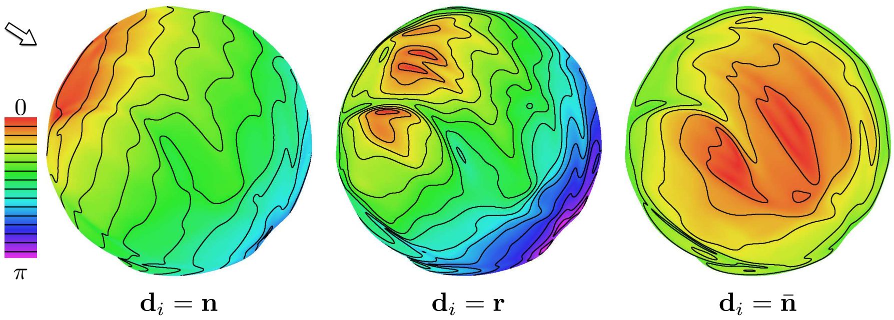

Figure 2: ***Shading parametrization****: Using a light direction oriented at top left with different parametrizations, a shading function achieves various behaviors on this blobby object. From left to right, we show the color code we used, then diffuse, specular and screen-aligned parametrizations.*

where $\mathbf{d}_i$ may take on different values depending on desired motion behavior: $\mathbf{d}_i = \mathbf{n}$ in the diffuse case and $\mathbf{d}_i = \mathbf{r}$ in the specular case, with $\mathbf{r}$ the mirrored view direction. We also found useful to introduce a screen-aligned parametrization, as with Lit Spheres, to create specific effects. In this case, the parametrization ignores the illumination reference frame orientation. This simple behavior is obtained by setting $\mathbf{d}_i = \bar{\mathbf{n}}$ with $\bar{\mathbf{n}}$ the normal expressed in screen-space. Figure 2 shows the three parametrizations.

The choice of profile function α_i controls the way shading intensity falls off away from $\mathbf{l}_i$ (or $u_i = 0$). We use a profile function that is a generalization of Phong shading:

$$\alpha_i(u) = \left(|\beta_i + (1-\beta_i)\cos(\zeta(u-\tau_i))|_+\right)^{\gamma_i} \tag{5}$$

where ζ clamps values to $[0,\pi]$ to ensure a monotonically decreasing function, and τ_i, β_i and γ_i are offset, bias and exponent parameters. Offset controls the angular portion of the parametrization where the shading function is equal to 1. Bias provides a control similar to the Half-Lambertian term in [GGSC98, MFE07]. Exponent determines the fall-off rate of shading, and is traditionally used in Phong's model to control material shininess.

In practice, the β_i and γ_i parameters are not easy to manipulate. We thus introduce alternative fall-off and cut-off controls (see Figure 3-left). The fall-off $f_i \geq 0$ controls the parametric location of the profile function inflection point, while the cut-off $c_i \geq f_i$ controls the parametric location at which the profile reaches 0. To evaluate u_i using these alternative controls, we set $\beta_i = \beta(c_i)$ and $\gamma_i = \gamma(f_i, c_i)$, as described in Equations 10 and 11 in Appendix A. With these parameters, artists directly control shading appearance with isocurve widgets (see Figure 3-right), instead of having to tune unintuitive parameters. Conventional shading profiles are easily reproduced with this function: Phong's diffuse term corresponds to $\tau_i = 0$, $f_i = c_i = \pi/2$; Half-lambertian is same as diffuse, except that $c_i = \pi$; Phong's specular term is same as diffuse, but leaves f_i free to vary in the $[0, \pi/2]$ range; cartoon shading occurs when $\tau_i = c_i$, and ambient shading when $\tau_i = \pi$ (used as a fill-in layer in our results).

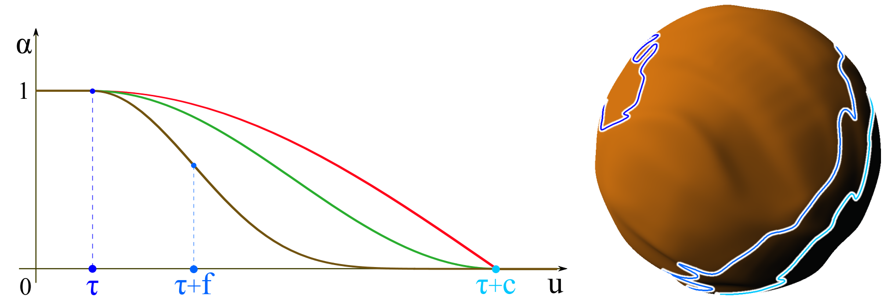

Figure 3: ***Shading profile****: our generalized Phong shading function is able to create a variety of styles. Left: The profile is controlled by intuitive offset, fall-off and cut-off parameters. The red, green and brown curves correspond to Lambert, Half-Lambert and custom profiles. Right: profile parameters are directly controlled on top of the object surface via isocurve widgets (in blue).*

3.3 Surface features

In traditional artistic and scientific illustrations, it is common to reveal surface features through the careful use of biased shading variations. *Mean Curvature Shading* [KWTM03] conveys shape details by assigning a pair of dark and bright colors to surface concavities and convexities respectively. Similarly, a pair of LitSpheres may be interpolated based on curvature information, as in the Z-Brush® software.

In our approach, we make use of surface curvature not only to modify shading color, but also the *extent* of shading shape. To this end, we first extract a measure of surface curvature κ from the depicted object. In our system, we use the view-centered mean curvature as defined in the work of Vergne et al. [VPB*09], because it provides automatic levels-of-detail and is computed dynamically from screen-space normals. However, our approach is independent of the choice of curvature measure. Taking into account surface shape turns a BADF into a 5D function:

$$\begin{array}{rccc} \rho_{\mathcal{A}}: & \mathbb{S}^2 \times \mathbb{S}^2 \times \mathbb{R} & \rightarrow & [0,1]^4 \\ & \mathbf{n}, \mathbf{v}, \kappa & \mapsto & (r,g,b,a). \end{array} \tag{6}$$

First, the color becomes a function of κ:

$$\begin{array}{rcl} \mathbf{c}_i(\kappa) & = & \lambda_i(\kappa)\hat{\mathbf{c}}_i + (1-\lambda_i(\kappa))\,\check{\mathbf{c}}_i \\ \lambda_i(\kappa) & = & 0.5 + 0.5\tanh(\kappa s_i) \end{array} \tag{7}$$

where $\hat{\mathbf{c}}_i$ and $\check{\mathbf{c}}_i$ are the colors assigned to convex and concave features respectively, $\lambda_i : \mathbb{R} \rightarrow [0,1]$ is a concave-to-convex transition function, and s_i is a user-controlled slope for this transition. Figure 4-left shows the effect obtained by using a pair of slightly different colors.

Second, the shading function is warped so that the light direction $\mathbf{l}_i$ is locally attracted to or repelled from $\mathbf{d}_i$. This warping is similar in idea to the Light Warping technique of Vergne et al. [VPB*09], except that we have an explicit knowledge of where the light direction comes from. This is done by locally altering the parametrization:

$$\begin{array}{rcl} \rho_i(\mathbf{n},\mathbf{v},\kappa) & = & \alpha_i \circ (u_i(\mathbf{n},\mathbf{v}) - T_i(\kappa)) \\ T_i(\kappa) & = & (2\lambda_i(\kappa) - 1)\,t_i \end{array} \tag{8}$$

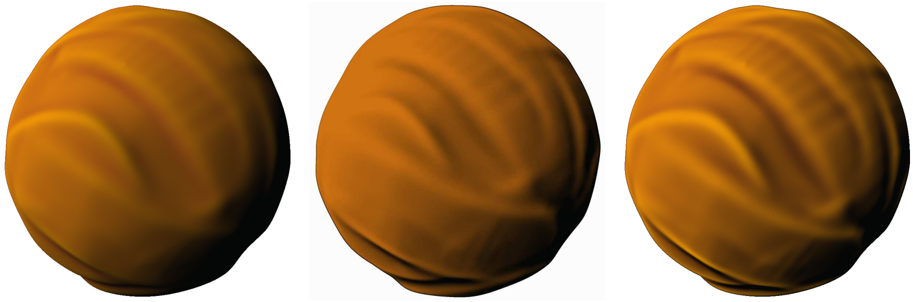

Figure 4: ***Shape depiction***: *we incorporate shape features into shading via two approaches. Left: we vary color in convexities and concavities as in Mean Curvature Shading. Middle: we modify shading extent in convex and concave regions. Right: applying both enhancement techniques.*

where $T_i : \mathbb{R} \rightarrow \mathbb{R}$ is a curvature-dependent translation in parametric space, and t_i is a user-controlled translation offset. Note that both $\mathbf{c}_i$ and ρ_i make use of the same concave-to-convex transition function; it ensures that shading color and shape variations are properly correlated, and makes manipulation easier for artists. Figure 4-middle shows the result of using shape features for manipulating shading only, while Figure 4-right shows the two methods in combination.

Augmented with these controls, our BADF model is able to depict shape features in a variety of ways, ranging from simple changes of color, to applying shading in concavities or convexities only, brightening convexities in otherwise dark regions (or darkening concavities in bright regions), or any other kind of combination.

3.4 Directional control

The last two sections described the behavior of each layer, and described a list of shading and shape parameters that we summarize in table 1.

blending function	b_i	*shading type*	$\mathbf{d}_i$
light direction	$\mathbf{l}_i$	shading offset	τ_i
shading fall-off	f_i	shading cut-off	c_i
shape slope	s_i	shape offset	t_i
convex color	$\hat{\mathbf{c}}_i$	concave color	$\check{\mathbf{c}}_i$

Table 1: *List of layer parameters*

Apart from the first two parameters in italics, all other parameters may be considered free variables, and could for instance be key-framed in an off-line animation system. In this paper, we rather focus on real-time stylized illustrations, and therefore we provide another type of control, based on *direction*. For artistic illustrations, we choose to control appearance based on lighting direction, expressed in screen-space. This choice makes it easy to finely control back-lighting effects in particular. In scientific illustrations, we rather use the

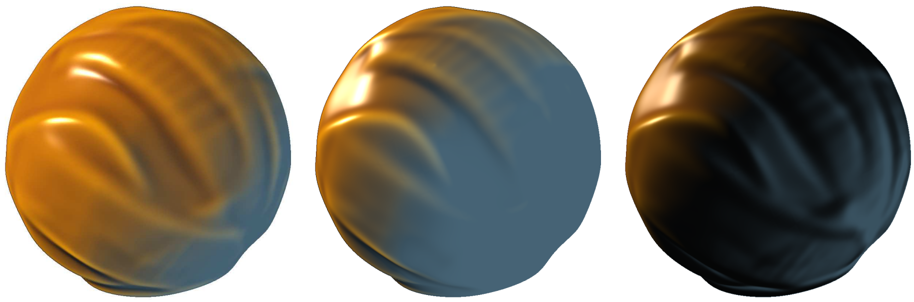

Figure 5: ***Directional control:*** *Shading parameters are easily varied per layer to interpolate between a few key directions. Left: the original BADF illuminated from top left. Middle: when lit from behind, back-light and highlights are too bright. Right: we slightly change the diffuse profile, darken the back-light and highlight colors, and extend the back-light toward concavities for this key lighting direction.*

direction of the view vector in world space, so that control may be based on the part of the object revealed from a given viewpoint. Other types of directional control might be used within our system, such as the orientation of the sun in outdoor 3D scenes, or objects proximity relations, but we leave their exploration to future work.

Our approach is to let an artist tune a set of BADF parameters for K key (lighting or viewing) directions ω_k; and then interpolate in-between to get a direction-dependent BADF. Figure 5 illustrates this functionality on a simple example. Consider an arbitrary parameter p_i for light $\mathbf{l}_i$. Its value is now given as a function of ω:

$$p_i(\omega) = \sum_{k=1}^{K} w_k(\omega) p_i^k \tag{9}$$

where p_i^k stores the parameter value in direction ω_k, and the w_k are weights ensuring a smooth interpolation and satisfying $p_i(\omega_k) = p_i^k$ and $\sum_k w_k = 1$.

We choose to use a biharmonic interpolation because it satisfies these requirements with C^2 continuity. Each key direction ω_k introduces a constraint on the sphere of directions, in the form of a weight vector $(w_1, w_2, \cdots, w_n)$, with all weights equal to 0 except for $w_k = 1$. We then solve for the biharmonic solution given these constraints using a least-squares minimization approach, as explained in Appendix B. Finally, we constrain the resulting weights to the $[0, 1]$ range with a smooth-step function, and a renormalization step. The resulting values interpolate key direction weights smoothly, and are well-adapted to our needs. However, we note that our shading technique is not dependent upon this choice of interpolation schemes, and other schemes like spherical radial basis functions [TS06] would work as well.

4 Results

We have implemented the layered BADF model described in the previous section mostly on the GPU, with the exception of the biharmonic weight interpolation.

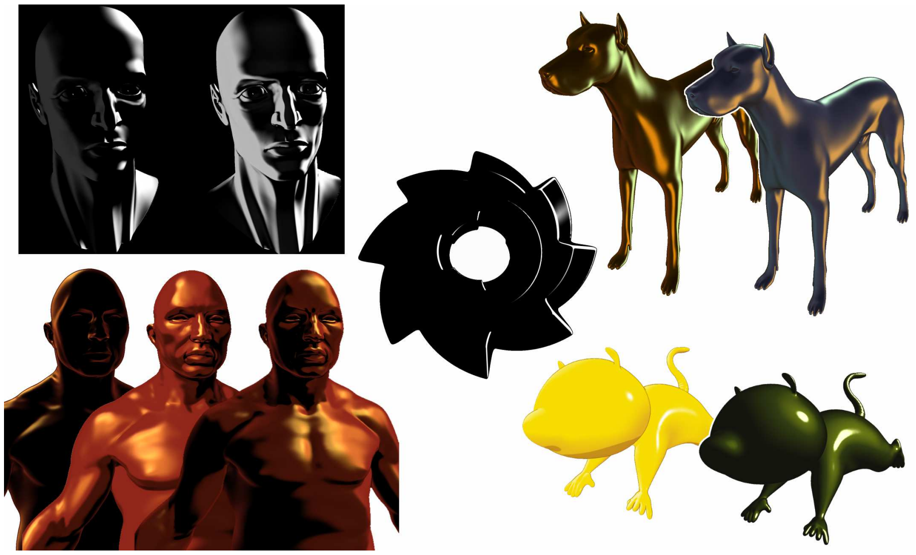

Figure 6: ***Artistic illustrations:*** *Our BADF model provides for a vast range of artistic shading styles, such as minimal lighting (center), dual lighting (top-left), comics-like (bottom-left), manga-like (bottom-right), and glossy effects (top-right).*

Our system runs in real-time ($\geq$ 30fps) on all our examples on a G80 graphics card in 1600×1200, which is adequate for BADF editing and creation. With our prototype system, we spent approximately 10 minutes on average for the creation of the BADFs seen in this paper. We also conducted a pilot user study, presented in supplementary material. It points out that despite the drawbacks of our UI, the system is easy to use and fairly intuitive, even for users without extensive experience designing materials. In the following, we present a panel of results obtained with our system for artistic and scientific illustration. A detailed collection of BADFs, including reference artwork, is given in supplementary materials and videos.

4.1 Artistic illustration

Figure 6 shows a series of results intended to reproduce artistic illustrations. The simplest one is shown at center. It is composed of only two layers: a black fill-in layer, and a white specular layer depicting convex shape features in a quite exaggerated way. Thanks to the warping technique, the resulting shading provides a compelling depiction of shape within a single layer. A pair of moderately more complex BADFs is shown at the top-left. In both cases, two opposing lights are used to depict all the details of the face. They only differ in their intensity and profile between the two figures.

The comics-like example at bottom-left of Figure 6 shows a character with a single BADF, lit from three different directions. Shape features are used to align shading with anatomical details and darken colors in concavities. Simpler BADFs, more akin to manga styles, are shown at bottom-right. Here, screen-aligned layers create effects such as contour lines and embossing. Glossy effects are easy to obtain as shown at top-right, where we illustrate two BADFs using a same number of layers, but with different colors and light directions. Here again, surface features help convey object shape regardless of chosen illumination.

4.2 Scientific illustration

Figure 7 demonstrates the subtle shading behaviors obtained with BADFs in scientific illustration scenarios. The pastel shading shown at top-left resembles sanguine drawings, and directional control permits to accurately position reddish and whitish shading functions around the hand silhouette when seen from behind. With a careful use of BADF layers, it also becomes possible to mimic existing materials as shown on the lemon and orange models at top-right.

In the bottom-left of Figure 7, we focused on reproducing the brilliant sheen of a beetle, and added directional control to enhance concavities when the specimen is seen from behind. Finally, we have combined three different BADFs in the image at bottom-right, in order to create a scientific illustration in symbolic colors. Here we make use of a pastel BADF for the bones to focus on internal organs, providing a neutral context.

Figure 7: ***Scientific illustrations**: Various kinds of scientific illustrations are simple to produce with our BADF model, including pastel sketches (top-left), mimicking real-world materials (top-right), creating a brilliant sheen in front while enhancing shapes from behind (bottom-left), and making use of symbolic and subdued colors (bottom-right).*

5 Discussion and future work

We designed our BADF model with a focus on interactive applications in mind. Its main idea is to encapsulate the behavior and effects of a few light sources to provide an integrated shading style in real-time, which is controlled at run-time by manipulating an illumination reference frame. This is adapted to interactive applications such as digital sculpture, scientific illustration, video games or even movie production (especially cartoons), in cases where end-users need not bother with the locations of individual light sources. The choice of directional lights follows this logic as it makes it easy to apply a same BADF to multiple 3D objects. Similarly, directional control permits richer variations without requiring any additional manipulation on the end-user side.

The extension to point light sources might not require many changes to our BADF model though. But the technique is also likely to lose some of its appeal since light positions would need to be adjusted for every new 3D object. In cases where a more advanced control on light positions is needed, a *nodal* BADF model might be more adapted. In such a scenario, each light source corresponds to a single node organized in a shading network, and is controlled independently. This is the conventional approach in movie production, and we believe our shading functions offer a contribution in this application domain as well. Artists might prefer to use key-framed instead of directional control in this case though.

There is room for improvements to our current BADF model. Shading functions are isotropic, and thus produce simple shading shapes. Although it is sufficient for creating a rich variety of shading styles, specific effects might require additional parameters. As shown in the supplemental materials, we experimented with simple transformations of the spherical parametrization, allowing us to reproduce more complex shading shapes such as in the work of Anjyo et al. [AH03, AWB06] or Pacanowski et al. [PGSP08]. However, we believe that such parametrized models are limited because they become less intuitive with an increasing number of parameters. In the future, we would thus like to investigate example-based approaches, enabling painting-like interactions in the vein of the work of Okabe et al. [OMSI07].

We presented two techniques to depict shape through shading, and combined them to produce shading styles that more resemble hand-made artistic and scientific illustrations. However, for extreme concave-to-convex transitions, both techniques tend to “break” the perception of material. This is explained by the fact that the gradient of shading is modified in a way that destroys basic material cues. A better understanding of material perception [Ade01] might give rise to more intuitive BADF models in the future. Moreover, we only considered opaque materials and ignored visibility. The extension to transparent and translucent materials or cast shadows might require novel parametrizations which take a 3D object’s volume into account. Again, understanding the perceptual cues triggered by such materials [FJB04] is likely to provide for simple and intuitive manipulations.

A Profile function

In this appendix, we give the formula for constructing the profile function introduced in Section 3.2. Without loss of generality, we consider the profile function

$$\eta(x) = \left(|\beta + (1-\beta)\cos(x)|_+\right)^\gamma$$

that is related to α by $\alpha(u) = \eta \circ \zeta(u - \tau)$, and where we have omitted the i subscript for clarity.

The cutoff parameter c corresponds to the angle at which the profile function reaches zero. The bias function $\beta(c)$ gives the bias value that satisfies this condition: $\eta(c) = 0$. It is defined as:

$$\beta(c) = -\frac{\cos(c)}{1 - \cos(c)} \tag{10}$$

The falloff parameter f marks the angle at which the profile function passes through an inflection point, which corresponds to the zero-crossing of its second derivative. The exponent function $\gamma(f, c)$ gives the (strictly positive) exponent value that satisfies this condition: $\eta''(f) = 0$. The profile's second derivative is:

$$\begin{aligned} \eta''(x) \quad = \quad & (\beta - 1)\gamma \cos x(\beta + (1 - \beta)\cos x)^{\gamma-1} + \\ & (1 - \beta)^2(\gamma - 1)\gamma(\beta + (1 - \beta)\cos x)^{\gamma-2} \sin^2 x \end{aligned}$$

There is only one strictly positive zero-crossing for $\eta''(f)$:

$$\gamma(f, c) = \frac{\beta - \beta\cos(f) - 1}{(\beta - 1)\sin^2(f)}$$

Replacing β by the formula for the bias function $\beta(c)$ gives:

$$\gamma(f, c) = \frac{1 - \cos(c)\cos(f)}{\sin^2(f)} \tag{11}$$

B Biharmonic interpolation

We detail our least-squares technique used to interpolate weights in Section 3.4. Our approximation consists in solving a discretized version of the bi-Laplace equation on a regularly tessellated sphere, which yields a biharmonic interpolation. It is an adaptation of a technique used for 2D animation [BBA09] from the planar to spherical domain. For each pair of adjacent triangles, the method minimizes the difference between the Jacobians of their vertex weights:

$$\arg\min_{\mathbf{w}} \sum_{a,b \in \mathcal{E}(T)} \|J_a(\mathbf{w}) - J_b(\mathbf{w})\|^2,$$

where $\mathcal{E}(T)$ is the set of triangle pairs that share an edge and $J_a(\mathbf{w})$ is the Jacobian of triangle a. The only difference with the planar approach is that we unfold adjacent triangles onto a common plane. This amounts to performing parallel transport of vectors along geodesics of the discrete sphere.

References

[Ade01] Adelson E. H.: On Seeing Stuff: The Perception of Materials by Humans and Machines. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series* (2001), vol. 4299, pp. 1–12.

[AH03] Anjyo K.-i., Hiramitsu K.: Stylized Highlights for Cartoon Rendering and Animation. *IEEE Computer Graphics and Applications 23*, 4 (2003), 54–61.

[AWB06] Anjyo K.-i., Wemler S., Baxter W.: Tweakable light and shade for cartoon animation. In *NPAR'2006: international symposium on non-photorealistic animation and rendering* (2006), ACM, pp. 133–139.

[BAOR06] Ben-Artzi A., Overbeck R., Ramamoorthi R.: Real-time BRDF editing in complex lighting. *ACM Trans. Graph.(Proc. SIGGRAPH 2006) 25*, 3 (2006), 945–954.

[BBA09] Baxter W., Barla P., Anjyo K.-i.: N-way morphing for 2D animation. *Computer Animation and Virtual Worlds (Proceedings of CASA) 20*, 2 (2009).

[BG07] Bruckner S., Gröller M. E.: Style transfer functions for illustrative volume rendering. *Computer Graphics Forum 26*, 3 (Sept. 2007), 715–724.

[BTM06] Barla P., Thollot J., Markosian L.: X-Toon: An extended toon shader. In *NPAR'2006: international symposium on non-photorealistic animation and rendering* (2006), ACM, pp. 127–132.

[CPK06] Colbert M., Pattanaik S., Krivanek J.: BRDF-Shop: Creating Physically Correct Bidirectional Reflectance Distribution Functions. *IEEE Computer Graphics and Applications 26*, 1 (2006), 30–36.

[EBJ*06] Edwards D., Boulos S., Johnson J., Shirley P., Ashikhmin M., Stark M., Wyman C.: The halfway vector disk for BRDF modeling. *ACM Transactions on Graphics 25*, 1 (2006), 1–18.

[FJB04] Fleming R. W., Jensen H. W., Bülthoff H. H.: Perceiving translucent materials. In *APGV '04: Proceedings of the 1st Symposium on Applied perception in graphics and visualization* (New York, NY, USA, 2004), ACM, pp. 127–134.

[GGSC98] Gooch A., Gooch B., Shirley P., Cohen E.: A non-photorealistic lighting model for automatic technical illustration. In *SIGGRAPH'98* (1998), ACM, pp. 447–452.

[Hod03] Hodges E. R. S.: *The Guild Handbook of Scientific Illustration.* Wiley, 2003.

[Hog91] Hogarth B.: *Dynamic Light and Shade.* Watson Guptill, 1991.

[KPC93] Kawai J. K., Painter J. S., Cohen M. F.: Radioptimization: goal based rendering. In *SIGGRAPH'93* (1993), ACM, pp. 147–154.

[KWTM03] Kindlmann G., Whitaker R., Tasdizen T., Möller T.: Curvature-Based Transfer Functions for Direct Volume Rendering: Methods and Applications. In *Proc. IEEE Visualization 2003* (October 2003), pp. 513–520.

[McC94] McCloud S.: *Understanding Comics: The Invisible Art.* Harper Paperbacks, 1994.

[MFE07] Mitchell J., Francke M., Eng D.: Illustrative rendering in Team Fortress 2. In *NPAR'2007: international symposium on non-photorealistic animation and rendering* (2007), ACM, pp. 71–76.

[OMSI07] Okabe M., Matsushita Y., Shen L., Igarashi T.: Illumination Brush: Interactive Design of All-Frequency Lighting. In *Pacific Conference on Computer Graphics and Applications* (2007), IEEE Computer Society, pp. 171–180.

[PGSP08] Pacanowski R., Granier X., Schlick C., Poulin P.: Sketch and Paint-based Interface for Highlight Modeling. In *Eurographics Workshop on Sketch-Based Interfaces and Modeling* (June 2008), Eurographics, pp. 17–23.

[RBD06] Rusinkiewicz S., Burns M., DeCarlo D.: Exaggerated Shading for Depicting Shape and Detail. *ACM Trans. Graph. (Proc. SIGGRAPH 2006) 25*, 3 (2006), 1199–1205.

[RH01] Ramamoorthi R., Hanrahan P.: A signal-processing framework for inverse rendering. In *SIGGRAPH'2001* (2001), ACM, pp. 117–128.

[SDS*93] Schoeneman C., Dorsey J., Smits B., Arvo J., Greenberg D.: Painting with light. In *SIGGRAPH'93* (1993), ACM, pp. 143–146.

[SMGG01] Sloan P.-P. J., Martin W., Gooch A., Gooch B.: The lit sphere: A model for capturing NPR shading from art. In *Graphics interface 2001* (2001), Canadian Information Processing Society, pp. 143–150.

[TABI07] Todo H., Anjyo K.-i., Baxter W., Igarashi T.: Locally controllable stylized shading. *ACM Trans. Graph.(Proc. SIGGRAPH 2007) 26*, 3 (2007), 17.

[TS06] Tsai Y.-T., Shih Z.-C.: All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers* (New York, NY, USA, 2006), ACM, pp. 967–976.

[VBGS08] Vergne R., Barla P., Granier X., Schlick C.: Apparent relief: A shape descriptor for stylized shading. In *NPAR '08: Proc. international symposium on Non-photorealistic animation and rendering* (2008), ACM, pp. 23–29.

[VPB*09] Vergne R., Pacanowski R., Barla P., Granier X., Schlick C.: Light warping for enhanced surface depiction. *ACM Trans. Graph. (Proc. SIGGRAPH 2009) 28*, 3 (Aug 2009), 1–8.

[Woo94] Wood P.: *Scientific Illustration: A Guide to Biological, Zoological, and Medical Rendering Techniques, Design, Printing, and Display*, 2nd ed. John Wiley and Sons, Inc., 1994.

Centre de recherche INRIA Bordeaux – Sud Ouest
Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex (France)

Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399